

# **Intel<sup>®</sup> Core<sup>™</sup> Duo Processor and Intel<sup>®</sup> Core<sup>™</sup> Solo Processor on 65 nm Process**

**Specification Update**

---

***April 2006***



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® Core™ Duo processor and Intel® Core™ Solo processor on 65 nm process may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel, Intel Core, Pentium, Celeron, Intel Xeon, Intel SpeedStep and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2006, Intel Corporation. All rights reserved.



# Contents

---

Revision History .....	4
Preface .....	5
Summary Tables of Changes .....	7
Identification Information .....	11
Errata .....	13
Specification Changes .....	25
Specification Clarifications .....	26
Documentation Changes .....	27



## Revision History

---

Revision	Description	Date
-001	Initial release	January 2006
-002	<ul style="list-style-type: none"><li>Updated Processor Identification (Table 1)</li></ul>	April 2006

§



## Preface

---

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

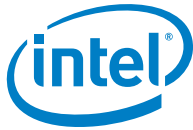
Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

### Affected Documents

Document Title	Document Number/Location
<i>The Intel® Core™ Duo Processor and the Intel® Core™ Solo Processor on 65 nm Process Datasheet</i>	<a href="#">309221-002</a>

### Related Documents

Document Title	Document Number/Location
<i>IA-32 Intel® Architecture Software Developer's Manual, Volume 1: Basic Architecture</i>	<a href="#">253665</a>
<i>IA-32 Intel® Architecture Software Developer's Manual, Volume 2A: Instruction Set Reference, A-M</i>	<a href="#">253666</a>
<i>IA-32 Intel® Architecture Software Developer's Manual, Volume 2B: Instruction Set Reference, N-Z</i>	<a href="#">253667</a>
<i>IA-32 Intel® Architecture Software Developer's Manual, Volume 3A: System Programming Guide</i>	<a href="#">253668</a>
<i>IA-32 Intel® Architecture Software Developer's Manual, Volume 3B: System Programming Guide</i>	<a href="#">253669</a>
<i>IA-32 Intel® Architecture Optimization Reference Manual</i>	<a href="#">248966</a>



## Nomenclature

**S-Spec Number** is a five-digit code used to identify products. Products are differentiated by their unique characteristics (e.g., core speed, L2 cache size, package type, etc.) as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number

**Errata** are design defects or errors. Errata may cause the Intel® Core™ Duo processor and the Intel® Core™ Solo processor on 65 nm process behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

**Specification Changes** are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

**Specification Clarifications** describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

**Documentation Changes** include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

**Note:** Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

§



## Summary Tables of Changes

---

The following table indicates the Specification Changes, Errata, Specification Clarifications or Documentation Changes, which apply to the listed MCH steppings. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

### Codes Used in Summary Table

#### Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

#### Status

Doc:	Document change or update that will be implemented.
PlanFix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Shaded:	This item is either new or modified from the previous version of the document.



**Note:** Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

- A = Intel® Pentium® II processor
- B = Mobile Intel® Pentium® II processor
- C = Intel® Celeron® processor
- D = Dual-Core Intel® Xeon™ Processor 2.80 GHz
- E = Intel® Pentium® III processor
- F = Intel® Pentium® processor Extreme Edition and Intel® Pentium® D processor
- G = Intel® Pentium® III Xeon™ processor
- H = Mobile Intel® Celeron® processor at 466 MHz, 433 MHz, 400 MHz, 366 MHz, 333 MHz, 300 MHz, and 266 MHz
- J = 64-bit Intel® Xeon™ processor MP with 1-MB L2 cache
- K = Mobile Intel® Pentium® III Processor - M
- L = Intel® Celeron® D Processor
- M = Mobile Intel® Celeron® processor
- N = Intel® Pentium® 4 processor
- O = Intel® Xeon™ processor MP
- P = Intel® Xeon™ processor
- Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading Technology on 90-nm process technology
- R = Intel® Pentium® 4 processor on 90 nm process
- S = 64-bit Intel® Xeon™ processor with 800 MHz system bus
- T = Mobile Intel® Pentium® 4 processor - M
- V = Mobile Intel® Celeron® processor on 0.13 micron process in micro-FCPGA package
- W = Intel® Celeron® M processor
- X = Intel® Pentium® M processor on 90 nm process with 2-MB L2 cache
- Y = Intel® Pentium® M processor
- Z = Mobile Intel® Pentium® 4 processor with 533 MHz system bus
- AA = Intel® Pentium® processor Extreme Edition and Intel® Pentium® D processor on 65 nm process
- AB = Intel® Pentium® 4 processor on 65 nm process
- AC = Intel® Celeron® processor in 478-Pin package
- AE = Intel® Core™ Duo processor and Intel® Core™ Solo processor on 65 nm process

Number	CO	Dual Core Only	Plans	ERRATA
AE1	X		No Fix	FST instruction with numeric and null segment exceptions may cause General Protection Faults to be missed and FP Linear Address (FLA) mismatch
AE2	X		No Fix	Code Segment limit violation may occur on 4-Gbyte limit check





Number	CO	Dual Core Only	Plans	ERRATA
AE3	X		No Fix	POPF and POPFD instructions that set the Trap Flag (TF) bit may cause unpredictable processor behavior
AE4	X		No Fix	REP MOVSB operation in fast string mode continues in that mode when crossing into a page with a different memory type
AE5	X		No Fix	Memory Aliasing with inconsistent A and D bits may cause processor deadlock
AE6	X		No Fix	VM Bit Will Be Cleared on a Double Fault Handler
AE7	X		No Fix	Page with PAT (Page Attribute Table) Set to USWC (Uncacheable Speculative Write Combine) while associated MTRR (Memory Type Range Register) Is UC (Uncacheable) may consolidate to UC
AE8	X		No Fix	FXSAVE after FNINIT without an Intervening FP (floating point) instruction may save uninitialized values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)
AE9	X		No Fix	Under certain conditions LTR (Load Task Register) instruction may result in system hang
AE10	X		No Fix	Invalid entries in Page-Directory-Pointer-Table Register (PDPTR) may cause General Protection (#GP) exception if the reserved bits are set to one
AE11	X		No Fix	REP MOVSB operation in fast string mode continues in that mode when crossing into a page with a different memory type
AE12	X		No Fix	FP inexact-result exception flag may not be set
AE13	X		No Fix	IFU/BSU deadlock may cause system hang
AE14	X		No Fix	MOV with debug register causes debug exception
AE15	X		No Fix	INIT does not clear global entries in the TLB
AE16	X		No Fix	Use of memory aliasing with inconsistent memory type may cause system hang
AE17	X		No Fix	Machine check exception may occur when interleaving code between different memory types
AE18	X		Plan Fix	Processor Digital Thermal Sensor (DTS) readout stops updating upon returning from C3/C4 State
AE19	X	X	No Fix	Data Prefetch Event Monitor (EMON) events can only be enabled on a single core
AE20	X		No Fix	LOCK# asserted during a special cycle shutdown transaction may unexpectedly deassert
AE21	X		No Fix	Disable execution-disable bit (IA32_MISC_ENABLES [34]) is shared between cores
AE22	X		No Fix	Last Branch Records (LBR) updates may be incorrect after a task switch



Number	CO	Dual Core Only	Plans	ERRATA
AE23	X		No Fix	Address reported by Machine-Check Architecture (MCA) on single-bit L2 ECC errors may be incorrect
AE24	X		No Fix	Disabling of single-step On Branch Operation may be delayed following a POPFD instruction
AE25	X		No Fix	Performance monitoring counters that count external bus events may report incorrect values after processor power state transitions
AE26	X		No Fix	VERW/VERR/LSL/LAR instructions may unexpectedly update the Last Exception Record (LER) MSR
AE27	X		No Fix	General Protection (#GP) fault may not be signaled on data segment limit violation above 4-G limit
AE28	X		No Fix	Performance Monitoring Events for retired floating point operations (C1h) may not be accurate
AE29	X		No Fix	DR3 address match on MOVD/MOVQ/MOVRTQ Memory store instruction may incorrectly increment performance monitoring count for saturating SIMD instructions executed (Event B1h)
AE30	X		No Fix	Global Pages in the Data Translation Look-Aside Buffer (DTLB) may not be flushed by RSM instruction before restoring the architectural state from SMRAM
AE31	X		No Fix	Data Breakpoint/Single Step on MOV SS/POP SS may be lost after entry into SMM
AE32	X		No Fix	CS limit violation on RSM may be serviced before higher priority Interrupts/Exceptions
AE33	X		No Fix	Hardware Prefetch Performance Monitoring Events may be counted inaccurately
AE34	X		No Fix	Pending x87 FPU Exceptions (#MF) following STI may be serviced before higher priority interrupts

Number	SPECIFICATION CHANGES
	There are no Specification Changes in this Specification Update revision.

Number	SPECIFICATION CLARIFICATIONS
	There are no Specification Clarifications in this Specification Update revision

Number	DOCUMENTATION CHANGES
	There are no Documentation Changes in this Specification Update revision



# Identification Information

## Component Identification via Programming Interface

The Intel® Core™ Duo processor and Intel® Core™ Solo processor on 65 nm process stepping can be identified by the following register contents:

Family <sup>1</sup>	Model <sup>2</sup>
0110	1110

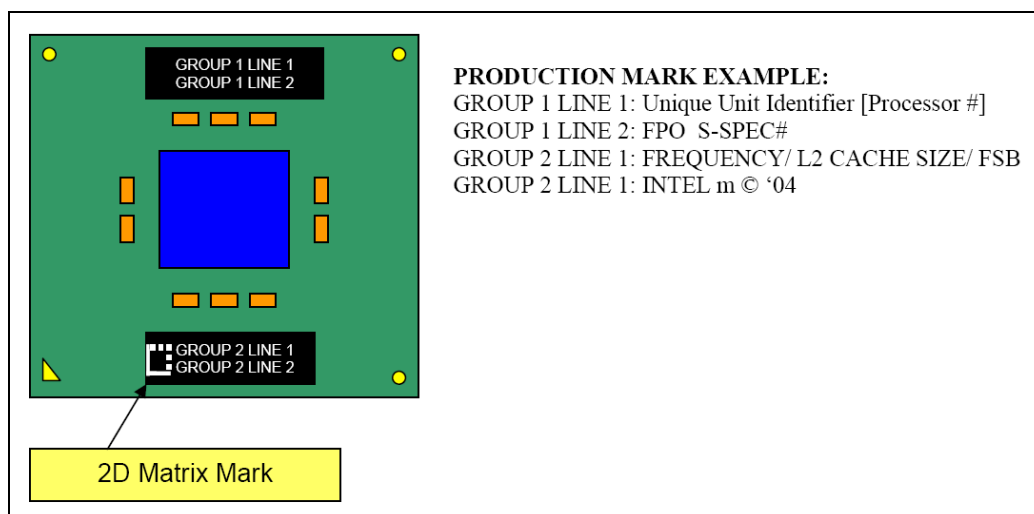
### NOTES:

1. The family corresponds to bit [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The family corresponds to bit [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.

Cache and TLB descriptor parameters are provided in the EAX, EBX, ECX, and EDX registers after the CUID instruction is executed with a 2 in the EAX register. Refer to the *Intel Processor Identification and the CUID Instruction Application Note (AP-485)* for further information on the CUID instruction.

## Component Marking Information

**Figure 1. Intel Core Duo Processor and Intel Core Solo Processor on 65 nm Process (Micro-FCPGA/FCBGA) S-Spec Markings**





**Table 1. Intel Core Duo Processor and Intel Core Solo Processor on 65 nm Process  
Identification Information**

QDF/S- SPEC#	Processor #	Package	Stepping	CPUID	FSB(MHz)	Speed HFM/LFM (GHz)	Notes
SL8W7	U1300	Micro-FCBGA	C-0	06E8h	533	1.06/0.80	4
SL8W6	U1400	Micro-FCBGA	C-0	06E8h	533	1.2/0.800	4
SL8VN	T2600	Micro-FCPGA	C-0	06E8h	667	2.16/1.00	2
SL8VP	T2500	Micro-FCPGA	C-0	06E8h	667	2.00/1.00	2
SL8VQ	T2400	Micro-FCPGA	C-0	06E8h	667	1.83/1.00	2
SL8VR	T2300	Micro-FCPGA	C-0	06E8h	667	1.66/1.00	2
SL8VS	T2600	Micro-FCBGA	C-0	06E8h	667	2.16/1.00	2
SL8VT	T2500	Micro-FCBGA	C-0	06E8h	667	2.00/1.00	2
SL8VU	T2400	Micro-FCBGA	C-0	06E8h	667	1.83/1.00	2
SL8VV	T2300	Micro-FCBGA	C-0	06E8h	667	1.66/1.00	2
SL92U	T1500	Micro-FCPGA	C-0	06E8h	667	2.00/1.00	1,2
SL92V	T1400	Micro-FCPGA	C-0	06E8h	667	1.83/1.00	1,2
SL8VY	T1300	Micro-FCPGA	C-0	06E8h	667	1.66/1.00	1,2
SL92W	T1500	Micro-FCBGA	C-0	06E8h	667	2.00/1.00	1,2
SL92X	T1400	Micro-FCBGA	C-0	06E8h	667	1.83/1.00	1,2
SL8W3	T1300	Micro-FCBGA	C-0	06E8h	667	1.66/1.00	1,2
SL8VW	L2400	Micro-FCBGA	C-0	06E8h	667	1.66/1.00	3
SL8VX	L2300	Micro-FCBGA	C-0	06E8h	667	1.50/1.00	3

**NOTES:**

1. Single Core Processor
2. VCC\_CORE=1.300-1.1625/1.000-0.7625 V for HFM/LFM Range; Deeper Sleep OVID Range=0.850-0.550 V; Deep C4 OVID Range=0.800-0.500 V.
3. VCC\_CORE=1.2125-1.000/1.000-0.7625 V for HFM/LFM Range; Deeper Sleep OVID Range=0.850-0.550 V; Deep C4 OVID Range=0.800-0.500 V.
4. VCC\_CORE=1.0500-0.950/0.9375 V for HFM range /LFM Range; Deeper Sleep OVID Range=0.800-0.750 V; Deep C4 OVID Range=0.750-0.650 V.



## Errata

---

### **AE1. FST instruction with numeric and null segment exceptions may cause General Protection Faults to be missed and FP Linear Address (FLA) mismatch**

**Problem:** FST instruction combined with numeric and null segment exceptions may cause General Protection Faults to be missed and FP Linear Address (FLA) mismatch.

**Implication:** This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

**Workaround:** None.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

### **AE2. Code Segment limit violation may occur on 4-Gbyte limit check**

**Problem:** Code Segment limit violation may occur on 4-Gbyte limit check when the code stream wraps around in a way that one instruction ends at the last byte of the segment and the next instruction begins at 0x0.

**Implication:** This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

**Workaround:** Avoid code that wraps around segment limit.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

### **AE3. POPF and POPFD instructions that set the Trap Flag (TF) bit may cause unpredictable processor behavior**

**Problem:** In some rare cases, POPF and POPFD instructions that set the Trap Flag (TF) bit in the EFLAGS register (causing the processor to enter Single-Step mode) may cause unpredictable processor behavior.

**Implication:** Single-Step operation is typically enabled during software debug activities, not during normal system operation.

**Workaround:** There is no workaround for Single-Step operation in commercially available software. For debug activities on custom software the POPF and POPFD instructions could be immediately followed by a NOP instruction to facilitate correct execution.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE4. REP MOVS operation in fast string mode continues in that mode when crossing into a page with a different memory type**

**Problem:** A fast “REP MOVS” operation continues to be handled in fast mode when the string operation crosses a page boundary into an Uncacheable (UC) memory type. Also if the fast string operation crosses a page boundary into a WC memory region, the processor does not self snoop the WC memory region. This may result in incorrect data for the WC portion of the operation if those cache lines were previously cached as WB (through aliasing) and modified.

**Implication:** String elements should be handled by the processor at the native operand size in UC memory. In the event that the WB to WC aliasing case occurs, the end result could vary— from benign software execution to operating system or application failure. Intel has not observed either aspects of this erratum in commercially available software.

**Workaround:** Software operating within Intel’s recommendation will not require WB and WC memory aliased to the same physical address.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE5. Memory aliasing with inconsistent A and D bits may cause processor deadlock**

**Problem:** In the event that software implements memory aliasing by having two Page Directory Entries (PDEs) point to a common Page Table Entry (PTE) and the Accessed and Dirty bits for the two PDEs are allowed to become inconsistent the processor may become deadlocked.

**Implication:** This erratum has not been observed with commercially available software.

**Workaround:** Software that needs to implement memory aliasing in this way should manage the consistency of the Accessed and Dirty bits.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE6. VM bit will be cleared on a Double Fault Handler**

**Problem:** Following a task switch to a Double Fault Handler that was initiated while the processor was in virtual-8086 (VM86) mode, the VM bit will be incorrectly cleared in EFLAGS.

**Implication:** When the OS recovers from the double fault handler, the processor will no longer be in VM86 mode.

**Workaround:** None.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).



**AE7. Page with PAT (Page Attribute Table) set to USWC (Uncacheable Speculative Write Combine) while associated MTRR (Memory Type Range Register) is UC (Uncacheable) may consolidate to UC**

**Problem:** A page whose PAT memory type is USWC while the relevant MTRR memory type is UC, the consolidated memory type may be treated as UC (rather than WC, as specified in *IA-32 Intel® Architecture Software Developer's Manual*).

**Implication:** When this erratum occurs, the memory page may be as UC (rather than WC). This may have a negative performance impact.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE8. FXSAVE after FNINIT without an intervening FP (floating point) instruction may save uninitialized values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)**

**Problem:** An FXSAVE after FNINIT without an intervening FP instruction may save uninitialized values for FDP and FDS.

**Implication:** When this erratum occurs, the values for FDP/FDS in the FXSAVE structure may appear to be random values. These values will be initialized by the first FP instruction executed after the FXRSTOR that restore the saved floating point state. Any FP instruction with memory operand will initialize FDP/FDS. Intel has not observed this erratum with any commercially available software.

**Workaround:** After an FNINIT, do not expect the FXSAVE memory image to be correct, until at least one FP instruction with a memory operand has been executed.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE9. Under certain conditions, LTR (Load Task Register) Instruction may result in system hang**

**Problem:** An LTR instruction may result in a system hang if all the following conditions are met:

1. Invalid data selector of the TR (Task Register) resulting with either #GP (General Protection Fault) or #NP (Segment Not Present Fault).
2. GDT (Global Descriptor Table) is not 8-bytes aligned.
3. Data BP (breakpoint) is set on cache line containing the descriptor data. When this erratum occurs, the memory page may be as UC (rather than WC). This may have a negative performance impact.

**Implication:** This erratum may result in system hang if all conditions have been met. This erratum has not been observed in commercial operating systems or software. For performance reasons, GDT is typically aligned to 8-bytes.

**Workaround:** Software should align GDT to 8-bytes.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).



**AE10. Invalid entries in Page-Directory-Pointer-Table Register (PDPTR) may cause General Protection (#GP) exception if the reserved bits are set to one**

**Problem:** Invalid entries in the Page-Directory-Pointer-Table Register (PDPTR) that have the reserved bits set to one may cause a General Protection (#GP) exception.

**Implication:** Intel has not observed this erratum with any commercially available software.

**Workaround:** Do not set the reserved bits to one when PDPTR entries are invalid.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE11. REP MOVS operation in fast string mode continues in that mode when crossing into a page with a different memory type**

**Problem:** A fast “REP MOVS” operation continues to be handled in fast mode when the string operation crosses a page boundary into an Uncacheable (UC) memory type. Also if the fast string operation crosses a page boundary into a WC memory region, the processor does not self snoop the WC memory region. This may result in incorrect data for the WC portion of the operation if those cache lines were previously cached as WB (through aliasing) and modified.

**Implication:** String elements should be handled by the processor at the native operand size in UC memory. In the event that the WB to WC aliasing case occurs, the end result could vary— from benign software execution to operating system or application failure. Intel has not observed either aspects of this erratum in commercially available software.

**Workaround:** Software operating within Intel’s recommendation will not require WB and WC memory aliased to the same physical address.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE12. FP inexact-result exception flag may not be set**

**Problem:** When the result of a floating-point operation is not exactly representable in the destination format (1/3 in binary form, for example), an inexact-result (precision) exception occurs. When this occurs, the PE bit (bit 5 of the FPU status word) is normally set by the processor. Under certain rare conditions, this bit may not be set when this rounding occurs. However, other actions taken by the processor (invoking the software exception handler if the exception is unmasked) are not affected. This erratum can only occur if the floating-point operation which causes the precision exception is immediately followed by one of the following instructions:

- FST m32real
- FST m64real
- FSTP m32real
- FSTP m64real
- FSTP m80real
- FIST m16int
- FIST m32int
- FISTP m16int
- FISTP m32int
- FISTP m64int





Note that even if this combination of instructions is encountered, there is also a dependency on the internal pipelining and execution state of both instructions in the processor.

**Implication:** Inexact-result exceptions are commonly masked or ignored by applications, as it happens frequently, and produces a rounded result acceptable to most applications. The PE bit of the FPU status word may not always be set upon receiving an inexact-result exception. Thus, if these exceptions are unmasked, a floating-point error exception handler may not recognize that a precision exception occurred. Note that this is a “sticky” bit, i.e., once set by an inexact-result condition, it remains set until cleared by software.

**Workaround:** This condition can be avoided by inserting two NOP instructions between the two floating-point instructions.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

#### **AE13. IFU/BSU deadlock may cause system hang**

**Problem:** A lockable instruction with memory operand that spans across two pages may, given some rare internal conditions, hang the system.

**Implication:** When this erratum occurs, the system may hang. Intel has not observed this erratum with any commercially available software or system.

**Workaround:** Lockable data should always be contained in a single page.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

#### **AE14. MOV with debug register causes debug exception**

**Problem:** When in V86 mode, if a MOV instruction is executed on debug registers, a general-protection exception (#GP) should be generated, as documented in the *Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Section 14.2. However, in the case when the general detect enable flag (GD) bit is set, the observed behavior is that a debug exception (#DB) is generated instead.

**Implication:** With debug-register protection enabled (i.e., the GD bit set), when attempting to execute a MOV on debug registers in V86 mode, a debug exception will be generated instead of the expected general-protection fault.

**Workaround:** In general, operating systems do not set the GD bit when they are in V86 mode. The GD bit is generally set and used by debuggers. The debug exception handler should check that the exception did not occur in V86 mode before continuing. If the exception did occur in V86 mode, the exception may be directed to the general-protection exception handler.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE15. INIT does not clear global entries in the TLB**

**Problem:** INIT may not flush a TLB entry when:

1. The processor is in protected mode with paging enabled and the page global enable flag is set (PGE bit of CR4 register)
2. G bit for the page table entry is set
3. TLB entry is present in TLB when INIT occurs.

**Implication:** Software may encounter unexpected page fault or incorrect address translation due to a TLB entry erroneously left in TLB after INIT.

**Workaround:** Write to CR3, CR4 or CR0 registers before writing to memory early in BIOS code to clear all the global entries from TLB.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE16. Use of memory aliasing with inconsistent memory type may cause system hang**

**Problem:** Software that implements memory aliasing by having more than one linear addresses mapped to the same physical page with different cache types may cause the system to hang. This would occur if one of the addresses is non-cacheable used in code segment and the other a cacheable address. If the cacheable address finds its way in instruction cache, and non-cacheable address is fetched in IFU, the processor may invalidate the non-cacheable address from the fetch unit. Any micro-architectural event that causes instruction restart will expect this instruction to still be in fetch unit and lack of it will cause system hang.

**Implication:** This erratum has not been observed with commercially available software.

**Workaround:** Although it is possible to have a single physical page mapped by two different linear addresses with different memory types, Intel has strongly discouraged this practice as it may lead to undefined results. Software that needs to implement memory aliasing should manage the memory type consistency.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE17. Machine check exception may occur when interleaving code between different memory types**

**Problem:** A small window of opportunity exists where code fetches interleaved between different memory types may cause a machine check exception. A complex set of micro-architectural boundary conditions is required to expose this window.

**Implication:** Interleaved instruction fetches between different memory types may result in a machine check exception. The system may hang if machine check exceptions are disabled. Intel has not observed the occurrence of this erratum while running commercially available applications or operating systems.

**Workaround:** Software can avoid this erratum by placing a serializing instruction between code fetches between different memory types.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE18. Processor Digital Thermal Sensor (DTS) Readout stops updating upon returning from C3/C4 state**

**Problem:** Digital Thermal Sensor (DTS) Readout is provided in IA32\_THERM\_STATUS bits 22:16. Upon waking up from C3/C4 low-power state, the DTS readout will no longer be updated.

**Implication:** Upon waking up from C3/C4 low-power state, software cannot rely on DTS readout any thermal threshold interrupts that are enabled in IA32\_THERM\_INTERRUPT, will also be affected.

**Workaround:** It is possible for BIOS to contain a workaround for this erratum.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE19. Data Prefetch Event Monitor (EMON) Events can only be enabled on a single core**

**Problem:** Current implementation of Data Prefetch EMON events allow counting only for a single core at a time.

**Implication:** Dual-core support for counting Data Prefetch EMON events is not currently available.

**Workaround:** Software should enable Data Prefetch EMON events on one core at a time.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE20. LOCK# asserted during a special cycle shutdown transaction may unexpectedly deassert**

**Problem:** During a processor shutdown transaction, when LOCK# is asserted and if a DEFER# is received during a snoop phase and the Locked transaction is pipelined on the front side bus (FSB), LOCK# may unexpectedly deassert.

**Implication:** When this erratum occurs, the system may hang during shutdown. Intel has not observed this erratum with any commercially available systems or software.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE21. Disable execution-disable bit (IA32\_MISC\_ENABLES [34]) is shared between cores**

**Problem:** The bit 34 of the IA32\_MISC\_ENABLES Model Specific Register (MSR) is shared between the execution cores.

**Implication:** Both cores will operate according to the shared value of bit IA32\_MISC\_ENABLES [34].

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE22. Last Branch Records (LBR) updates may be incorrect after a task switch**

**Problem:** A Task-State Segment (TSS) task switch may incorrectly set the LBR\_FROM value to the LBR\_TO value.

**Implication:** The LBR\_FROM will have the incorrect address of the Branch Instruction.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE23. Address reported by Machine-Check Architecture (MCA) on single-bit L2 ECC errors may be incorrect**

**Problem:** When correctable single-bit ECC errors occur in the L2 cache the address is logged in the MCA address register (MCI\_ADDR). Under some scenarios, the address reported may be incorrect.

**Implication:** Software should not rely on the value reported in MCI\_ADDR, for Single-bit L2 ECC errors

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE24. Disabling of single-step On Branch Operation may be delayed following a POPFD instruction**

**Problem:** Disabling of Single-step On Branch Operation may be delayed, if the following conditions are met:

1. "Single Step On Branch Mode" is enabled (DebugCtlMSR.BTF and EFLAGS.TF are set)
2. POPFD used to clear EFLAGS.TF
3. A jump instruction (JMP, Jcc, etc.) is executed immediately after POPFD

**Implication:** Single-step On Branch mode may remain in effect for one instruction after the POPFD instruction disables it by clearing the EFLAGS.TF bit.

**Workaround:** There is no workaround for Single-Step operation in commercially available software. The workaround for custom software is to execute at least one instruction following POPFD before issuing a JMP instruction.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE25. Performance monitoring counters that count external bus events may report incorrect values after processor power state transitions**

**Problem:** Performance monitoring counters that count external bus events operate when the processor is in the Active state (C0). If a processor transitions to a new power state, these Performance monitoring counters will stop counting, even if the event being counted remains active.

**Implication:** After transitioning between processor power states, software may observe incorrect counts in Performance monitoring counters that count external bus events.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE26. VERW/VERR/LSL/LAR instructions may unexpectedly update the Last Exception Record (LER) MSR**

**Problem:** The LER MSR may be unexpectedly updated, if the resultant value of the Zero Flag (ZF) is zero after executing the following instructions:

1. VERR (ZF=0 indicates unsuccessful segment read verification)
2. VERW (ZF=0 indicates unsuccessful segment write verification)
3. LAR (ZF=0 indicates unsuccessful access rights load)
4. LSL (ZF=0 indicates unsuccessful segment limit load)

**Implication:** The value of the LER MSR may be inaccurate if VERW/VERR/LSL/LAR instructions are executed after the occurrence of an exception.

**Workaround:** Software exception handlers that rely on the LER MSR value should read the LER MSR before executing VERW/VERR/LSL/LAR instructions.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE27. General Protection (#GP) fault may not be signaled on data segment limit violation above 4-G limit**

**Problem:** Memory accesses to flat data segments (base = 00000000h) that occur above the 4-G limit (0ffffffffh) may not signal a #GP fault.

**Implication:** When such memory accesses occur, the system may not issue a #GP fault.

**Workaround:** Software should ensure that memory accesses do not occur above the 4-G limit (0xffffffffh).

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE28. Performance Monitoring Events for retired floating point operations (C1h) may not be accurate**

**Problem:** Performance Monitoring Events that count retired floating point operations may be too high.

**Implication:** The Performance Monitoring Event may have an inaccurate count.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE29. DR3 address match on MOVD/MOVQ/MOVNTQ memory store instruction may incorrectly increment performance monitoring count for saturating SIMD instructions executed (Event B1h)**

**Problem:** Performance monitoring for Event B1h normally increments on saturating SIMD instruction executed. Regardless of DR7 programming, if the linear address of a memory store MOVD/MOVQ/MOVNTQ instruction executed matches the address in DR3, the B1h counter may be incorrectly incremented.

**Implication:** The value observed for performance monitoring count for saturating SIMD instructions executed may be too high.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE30. Global Pages in the Data Translation Look-Aside Buffer (DTLB) may not be flushed by RSM instruction before restoring the architectural state from SMRAM**

**Problem:** Resume from System Management Mode (RSM) does not flush global pages from DTLB before the System Management RAM (SMRAM) loads.

**Implication:** If SMM turns on paging with global paging enabled and then maps any of linear addresses of SMRAM using global pages, RSM load may load data from the wrong location.

**Workaround:** Do not use global pages in system management mode.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE31. Data Breakpoint/Single Step on MOV SS/POP SS may be lost after entry into SMM**

**Problem:** Data Breakpoint/Single Step exceptions are normally blocked for one instruction following MOV SS/POP SS instructions. Immediately after executing these instructions, if the processor enters SMM (System Management Mode), upon RSM (resume from SMM) operation, normal processing of Data Breakpoint/Single Step exceptions is restored.

Because of this erratum, Data Breakpoints/Single step exceptions on MOVSS/POPSS instructions may be lost under one of the following conditions.

1. Following SMM entry and after RSM, the next instruction to be executed is HLT or MWAIT
2. SMM entry after executing MOV SS/POP SS is the result of executing an I/O instruction that triggers a synchronous SMI (System Management Interrupt).

**Implication:** Data Breakpoints/Single step operation on MOV SS/POP SS instructions may be unreliable in the presence of SMIs.

**Workaround:** None Identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE32. CS limit violation on RSM may be serviced before higher priority Interrupts/Exceptions**

**Problem:** When the processor encounters a CS (Code Segment) limit violation, a #GP (General Protection Exception) fault is generated after all higher priority Interrupts and Exceptions are serviced. Because of this erratum, if RSM (Resume from System Management Mode) returns to execution flow where a CS limit violation occurs, the #GP fault may be serviced before a higher priority Interrupt or Exception (e.g., NMI (Non-Maskable Interrupt), Debug break(#DB), Machine Check (#MC), etc.).

**Implication:** Operating systems may observe a #GP fault being serviced before higher priority Interrupts and Exceptions.

**Workaround:** None Identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

**AE33. Hardware Prefetch Performance Monitoring Events may be counted inaccurately**

**Problem:** Hardware prefetch activity is not accurately reflected in the Hardware Prefetch Performance Monitoring.

**Implication:** This erratum may cause inaccurate counting for all Hardware Prefetch Performance Monitoring Events.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).



**AE34. Pending x87 FPU Exceptions (#MF) following STI may be serviced before higher priority interrupts**

**Problem:** Interrupts that are pending prior to the execution of the STI (Set Interrupt Flag) instruction are serviced immediately after the STI instruction is executed. Because of this erratum, if following STI, an instruction that triggers a #MF is executed while STPCLK#, Enhanced Intel SpeedStep® Technology transitions or Intel® Thermal Monitor 1 events occur, the pending #MF may be serviced before higher priority interrupts.

**Implication:** Software may observe #MF being serviced before higher priority interrupts.

**Workaround:** None identified.

**Status:** For the steppings affected, see the [Summary Tables of Changes](#).

§





## *Specification Changes*

---

There are no specification changes in this specification update revision.

**Note:** All specification changes will be incorporated into a future version of the *Intel® Core™ Duo Processor and Intel® Core™ Solo Processor on 65 nm Process Datasheet*.

§



## *Specification Clarifications*

---

There are no specification clarifications in this specification update revision.

**Note:** All specification changes will be incorporated into a future version of the *Intel® Core™ Duo Processor and Intel® Core™ Solo Processor on 65 nm Process Datasheet*.

§



## Documentation Changes

---

There are no documentation changes in this specification update revision.

- 

**Note:** All specification changes will be incorporated into a future version of the appropriate Intel® Core™ Duo Processor and Intel® Core™ Solo Processor on 65 nm Process Datasheet.

**Note:** Documentation changes for *IA-32 Intel® Architecture Software Developer's Manuals volumes 1, 2A, 2B, 3A and 3B* will be posted in a separate document *IA-32 Intel® Architecture and Intel® Extended Memory 64 Technology Software Developer's Manual Documentation Changes*. Follow the link below to become familiar with this file.

<http://developer.intel.com/design/pentium4/specupdt/252046.htm>